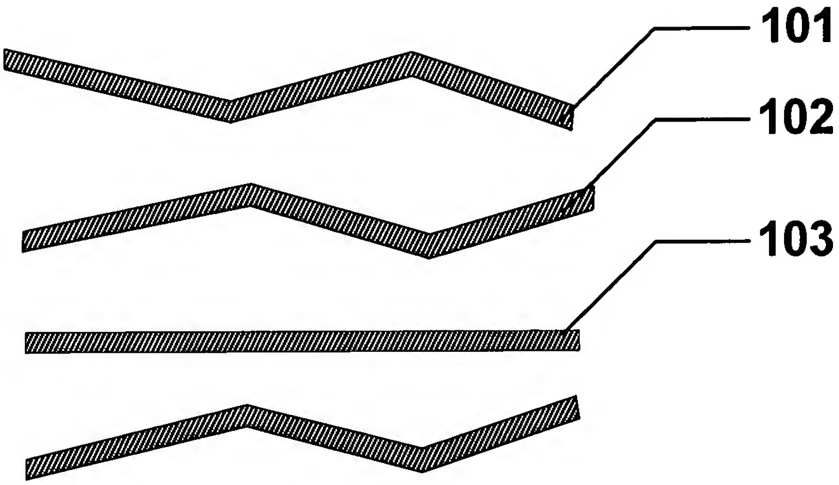

Figure 1A

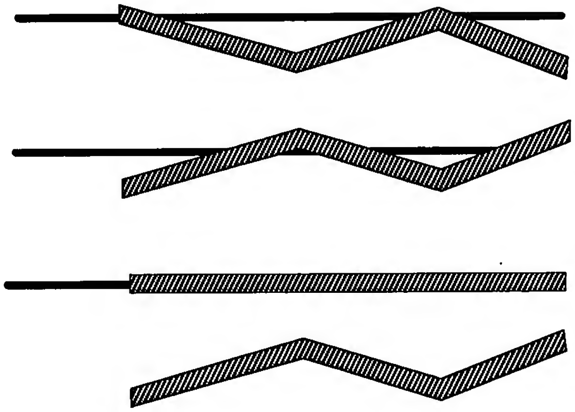

Figure 1B

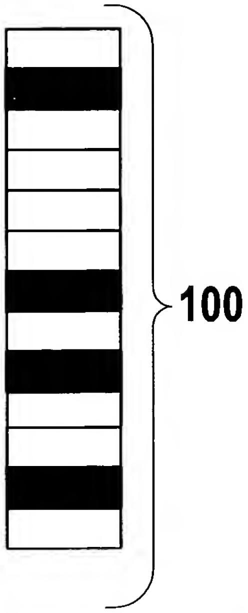

Figure 2A

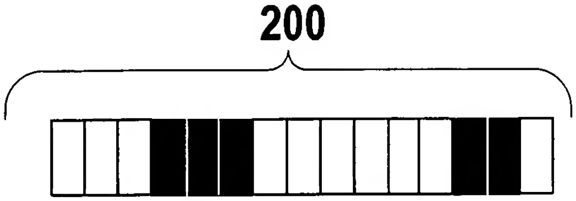

Figure 2B

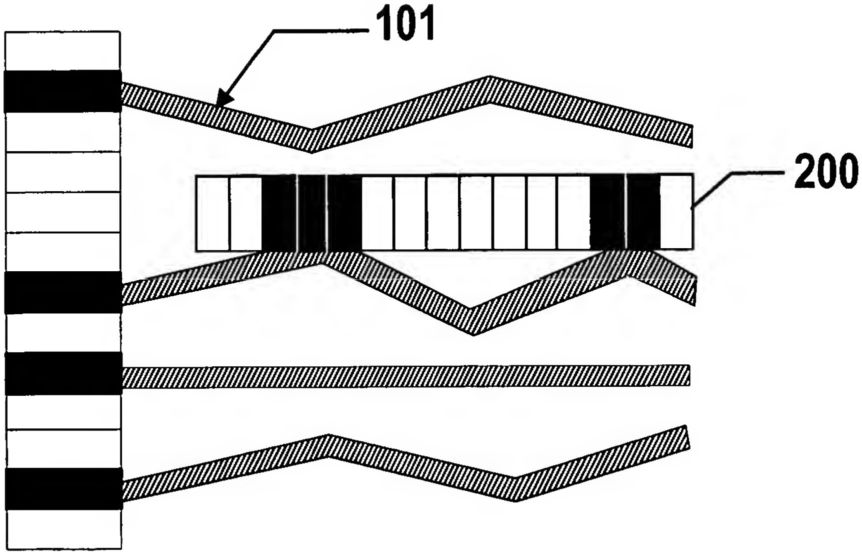

Figure 3

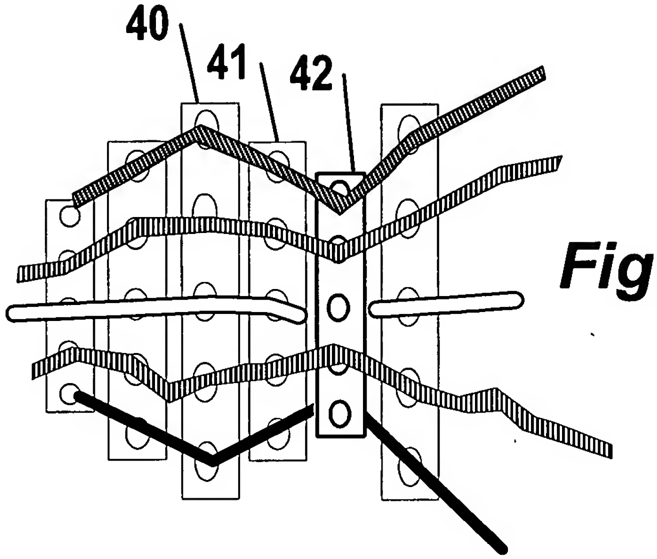

Figure 4A

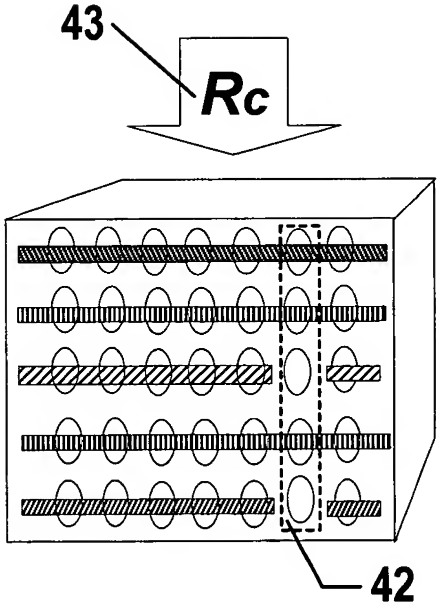

Figure 4B

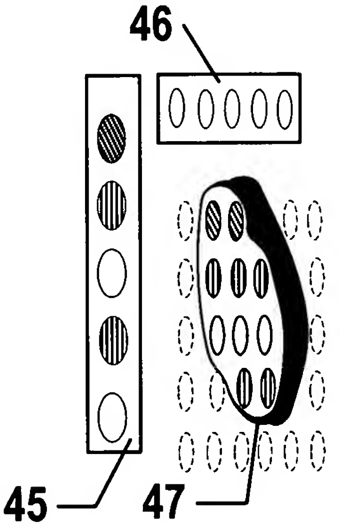

Figure 4C

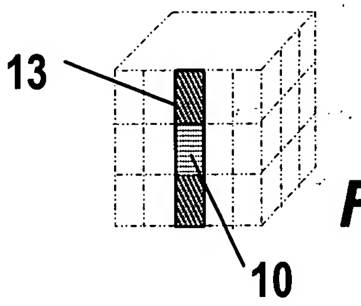

Figure 5

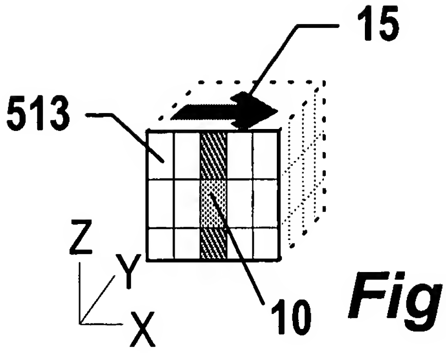

Figure 6

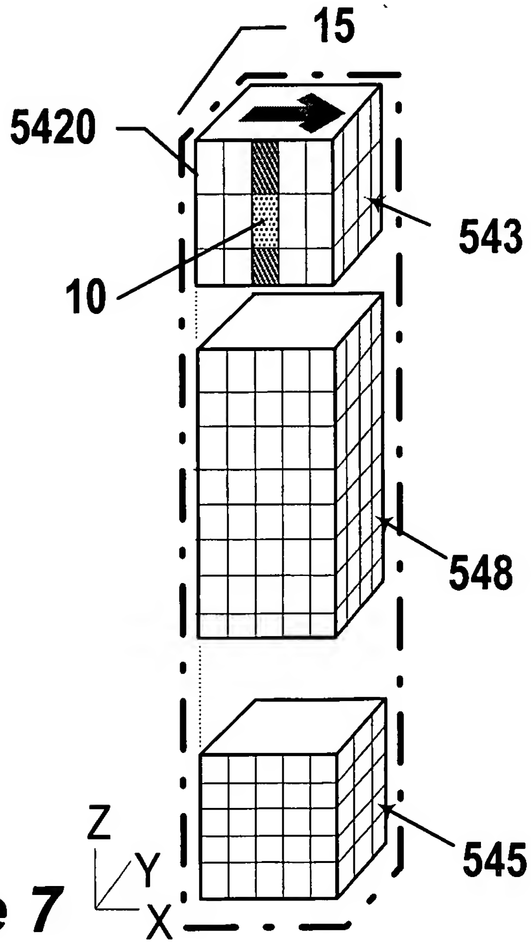

Figure 7

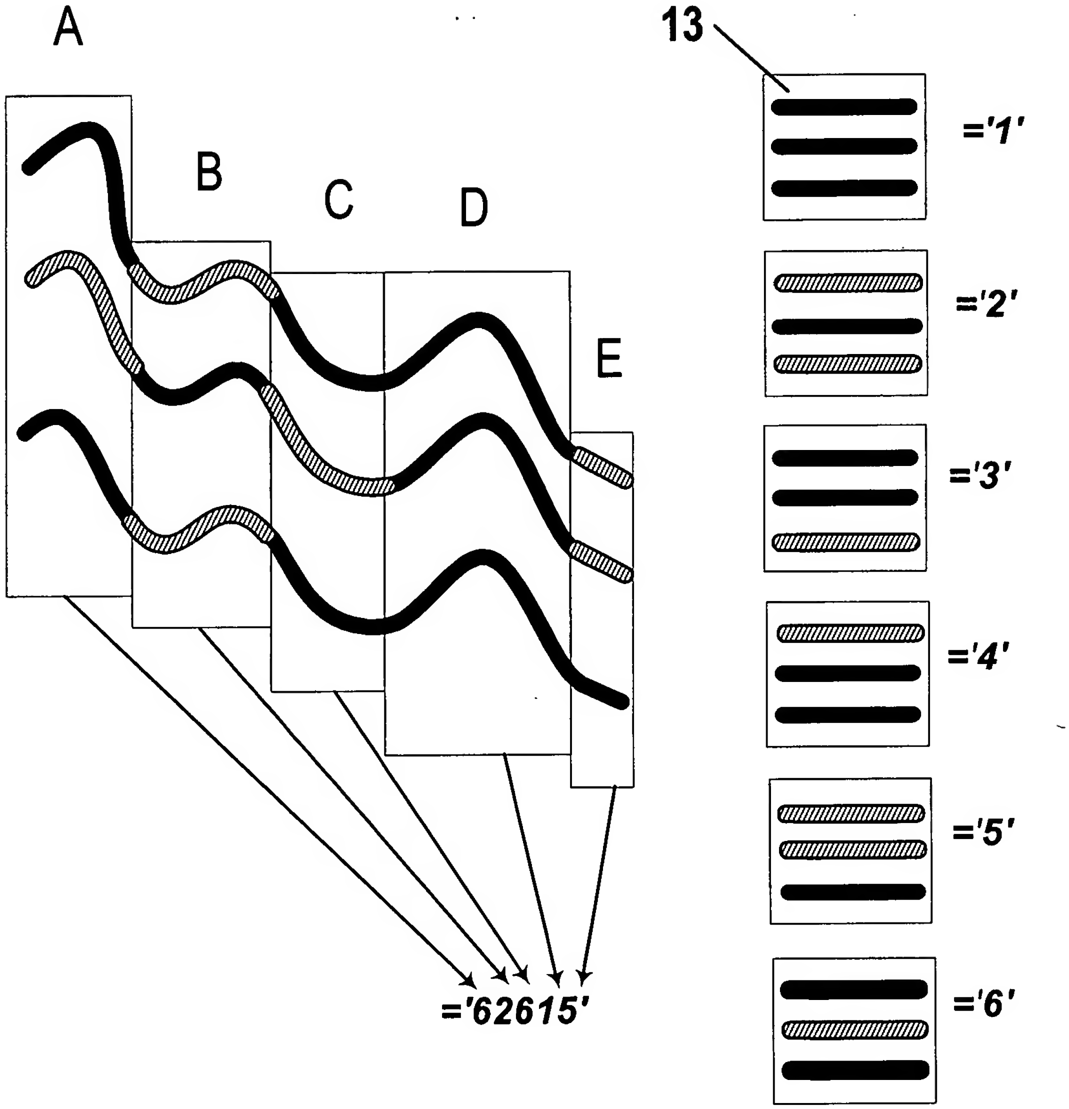

Figure 8

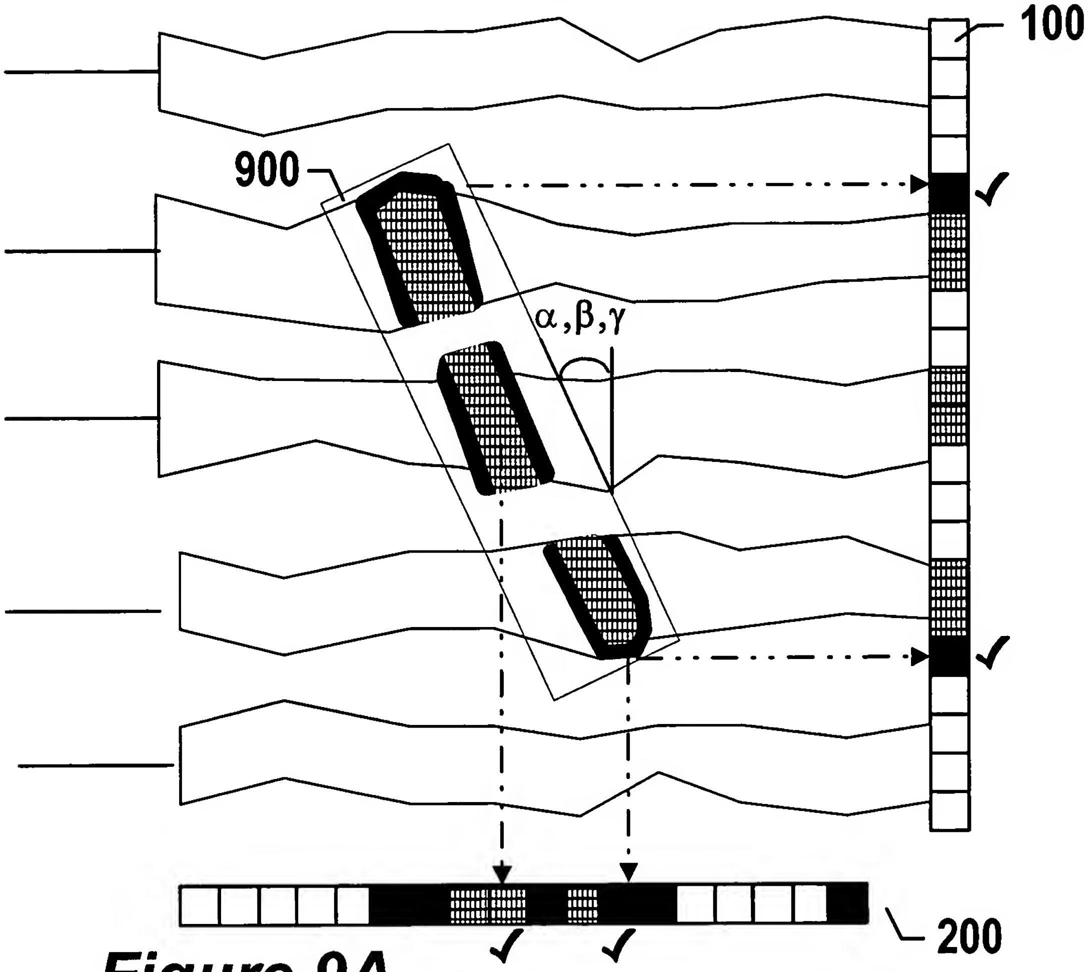

Figure 9A

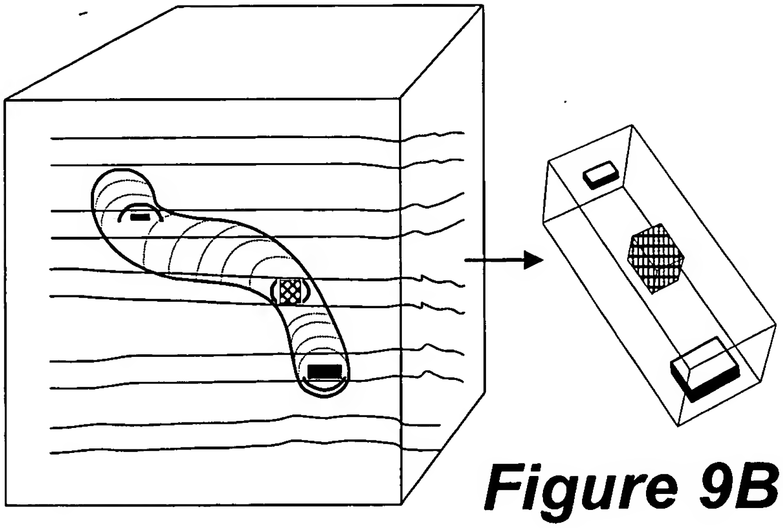

Figure 9B

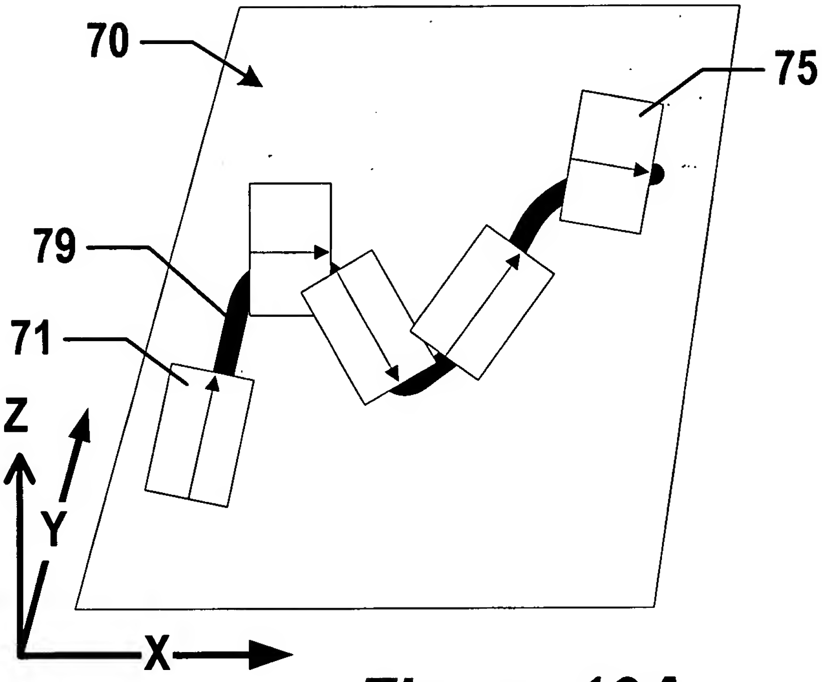

Figure 10A

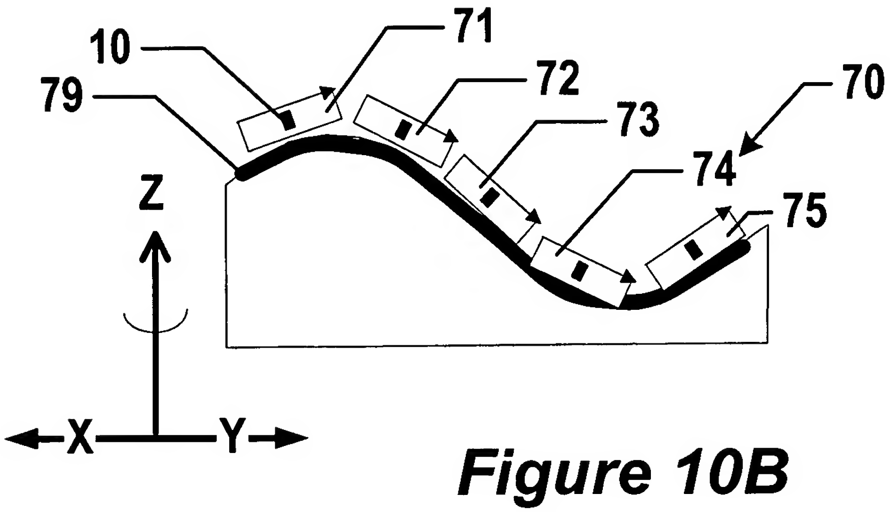

Figure 10B

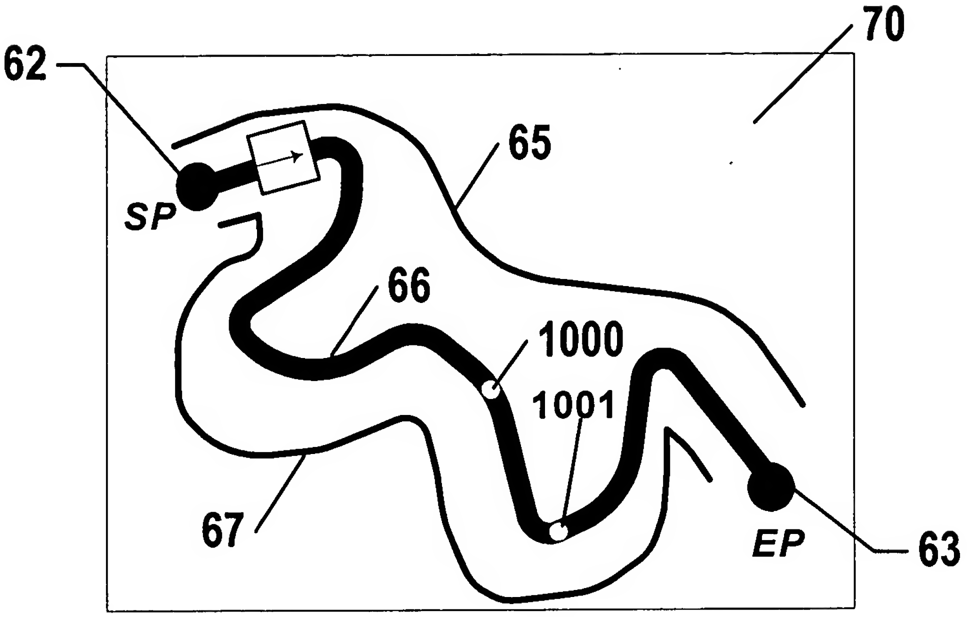

Figure 11A

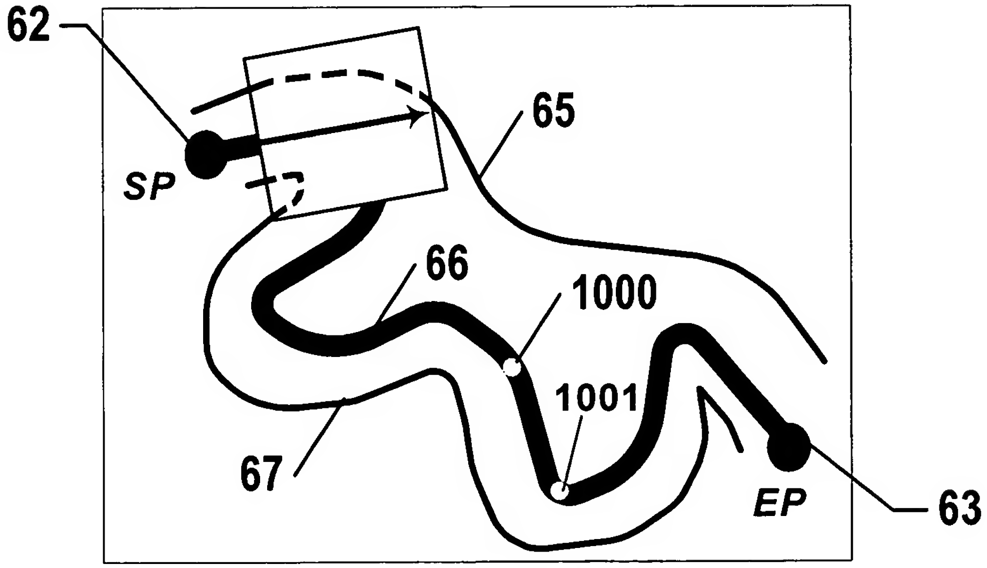

Figure 11B

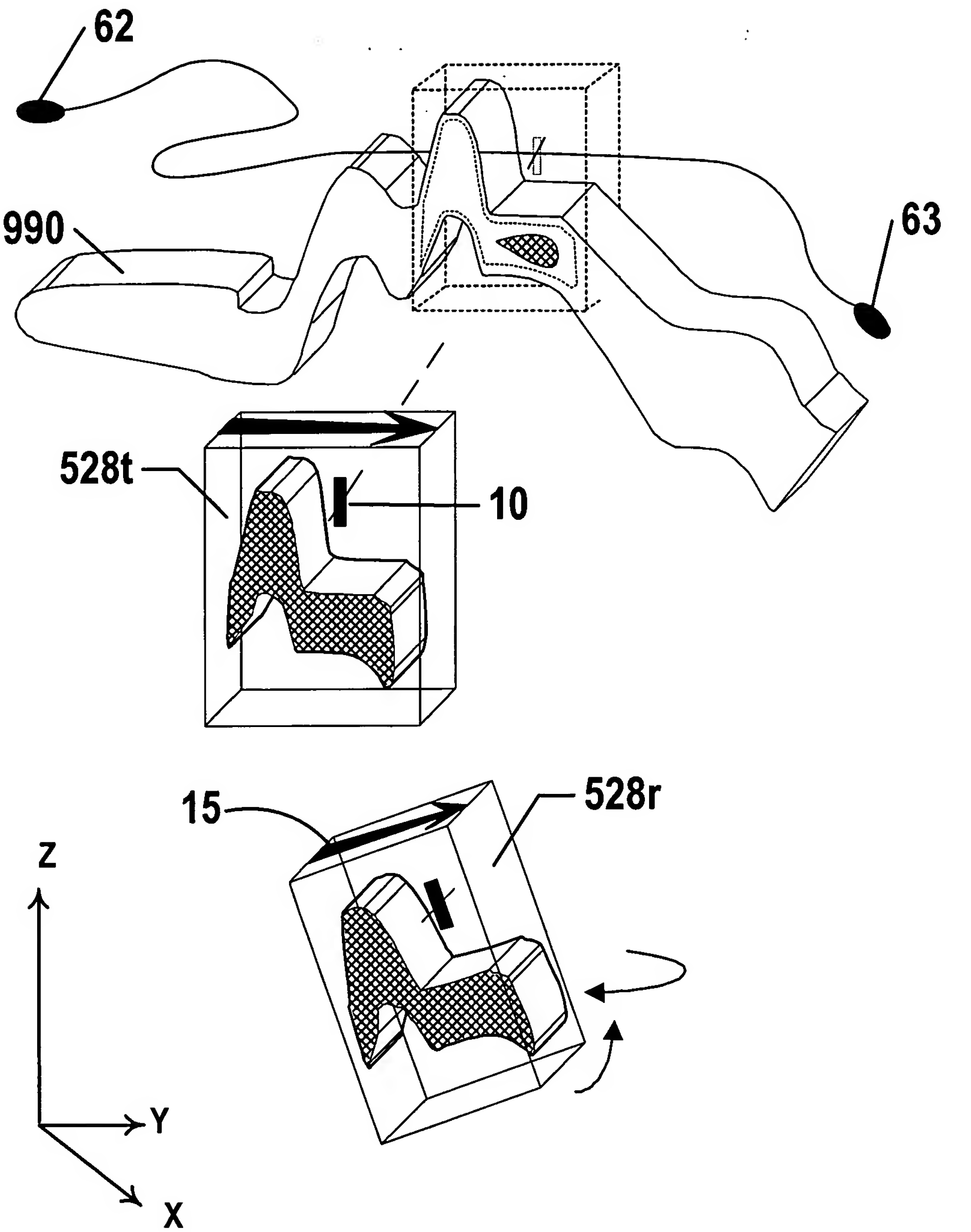

Figure 12

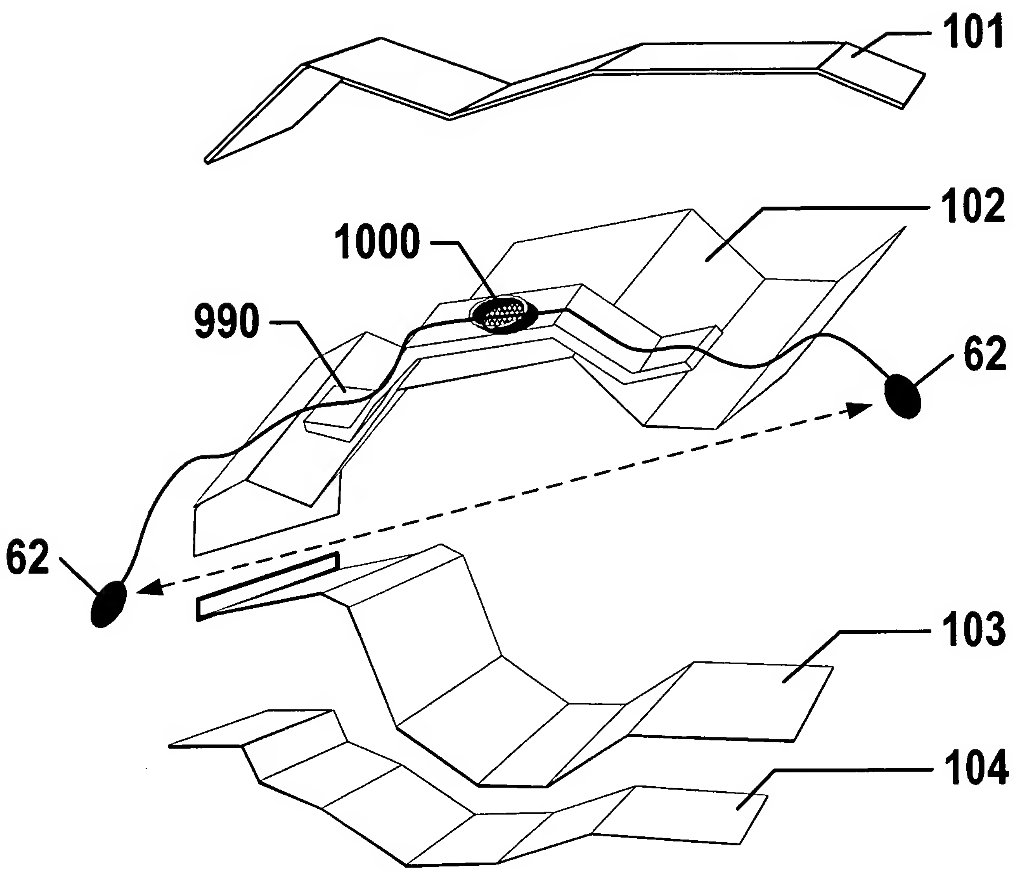

Figure 13

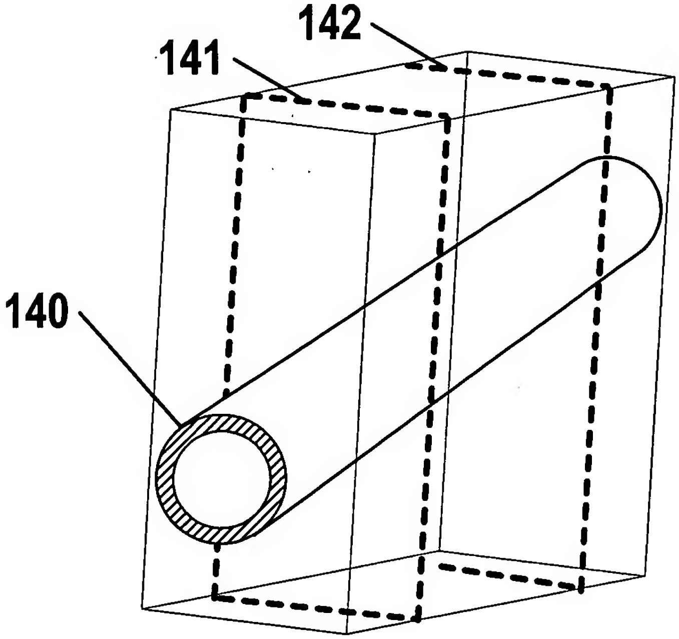

Figure 14A

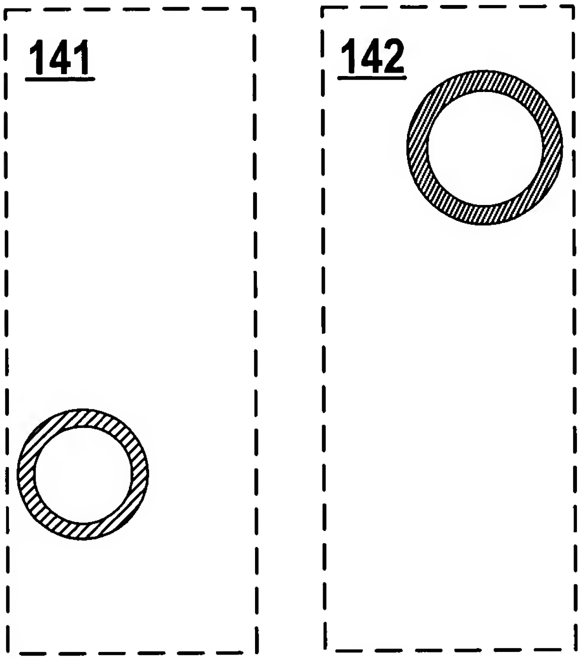

Figure 14B

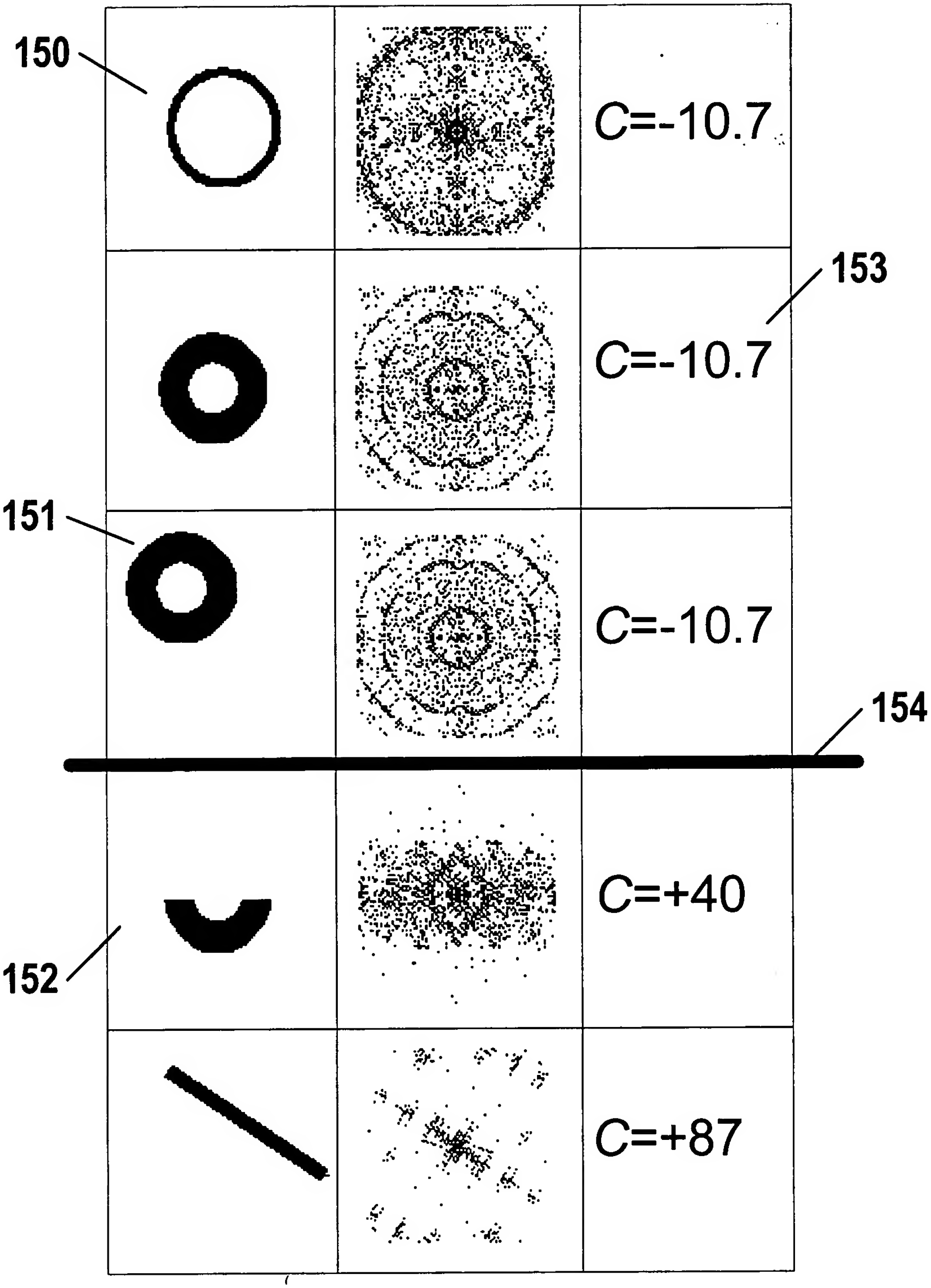

Figure 15

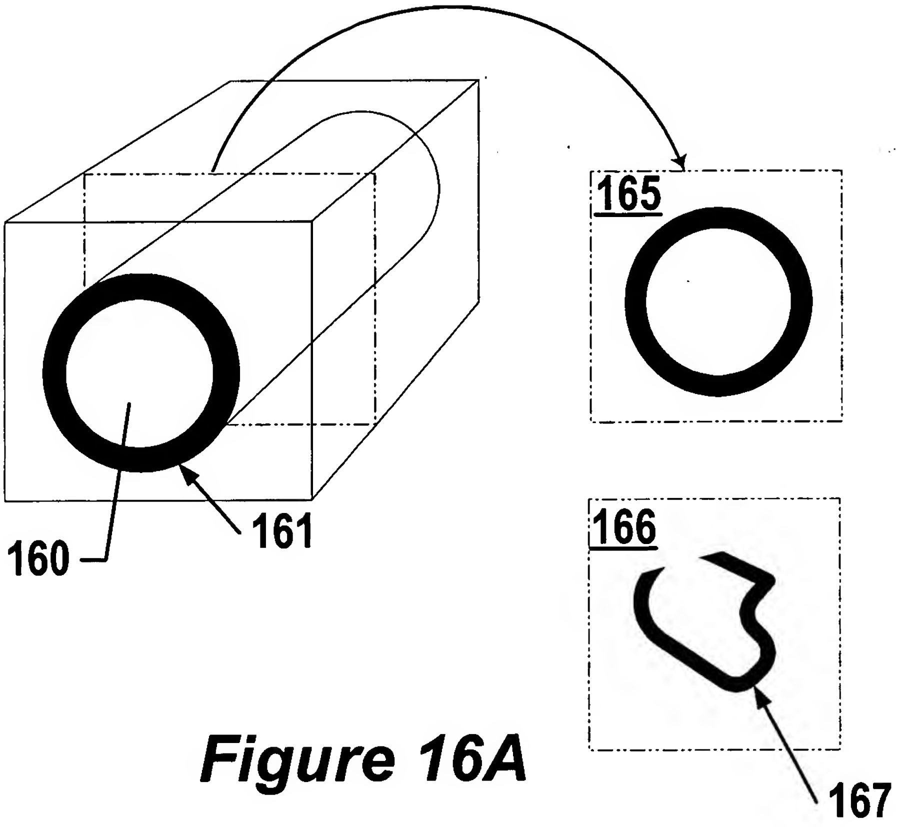

Figure 16A

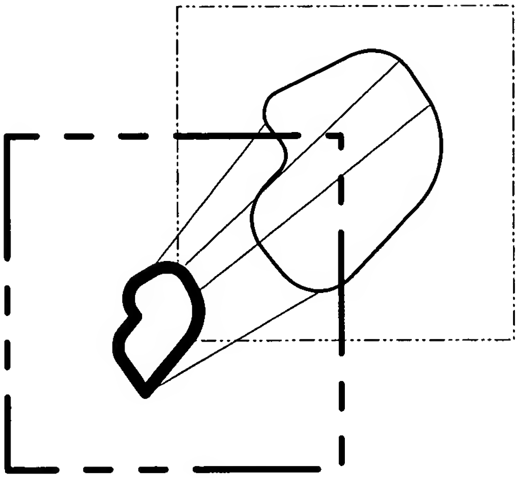

Figure 16B

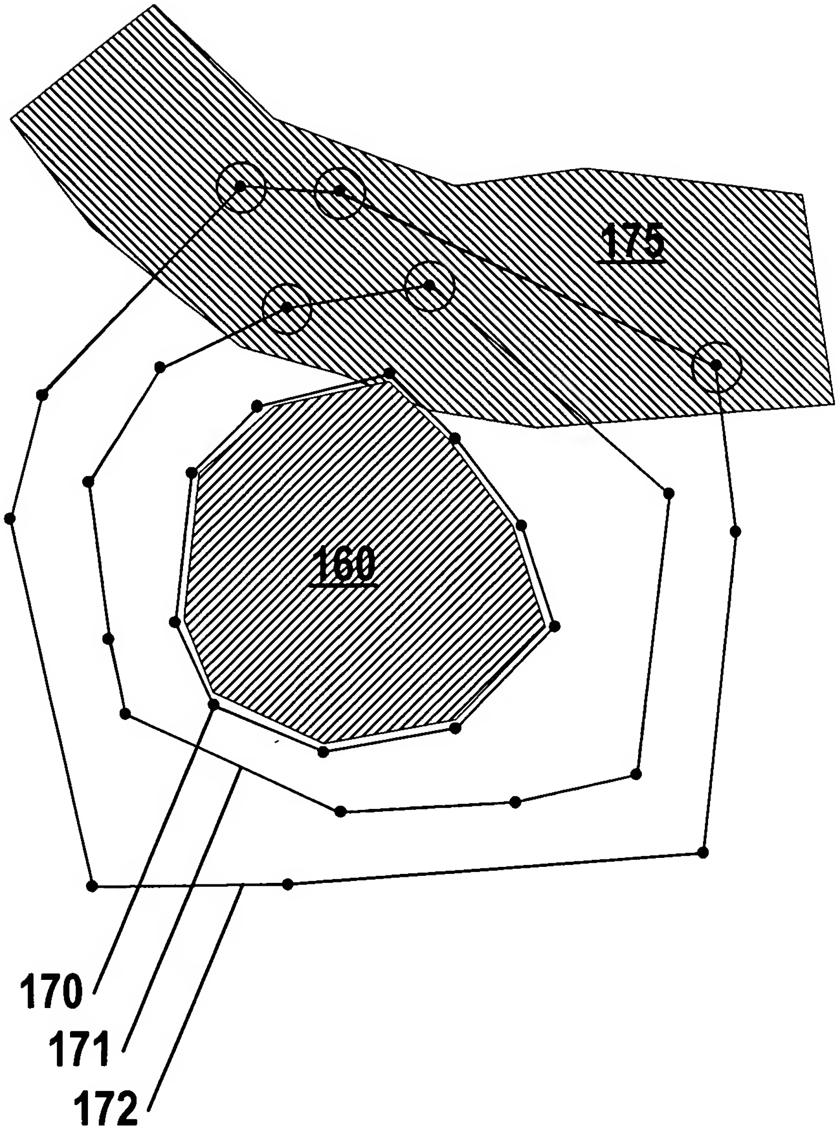

Figure 17

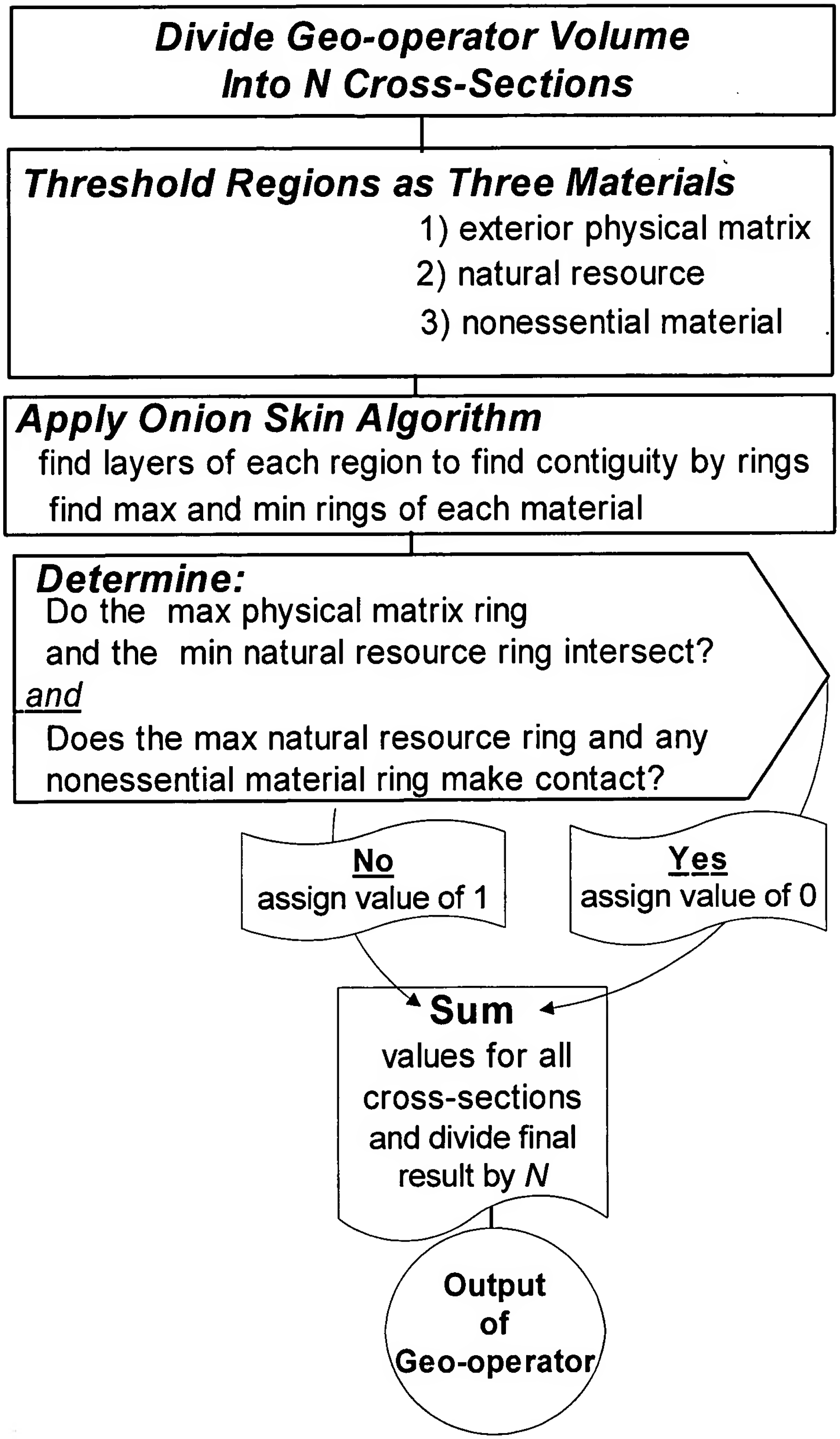
Divide Geo-operator Volume Into N Cross-Sections
Threshold Regions as Three Materials
1) exterior physical matrix
2) natural resource
3) nonessential material
Apply Onion Skin Algorithm
find layers of each region to find contiguity by rings
find max and min rings of each material
Determine:
Do the max physical matrix ring and the min natural resource ring intersect?
and
Does the max natural resource ring and any nonessential material ring make contact?
No
assign value of 1
Yes
assign value of 0
Sum
values for all cross-sections and divide final result by N
Output of Geo-operator

Figure 18

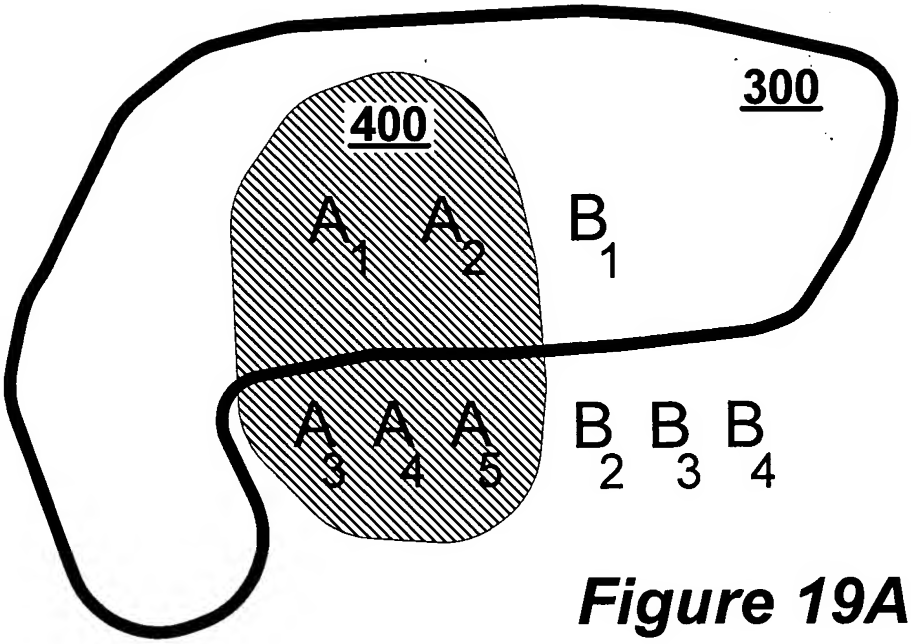

Figure 19A

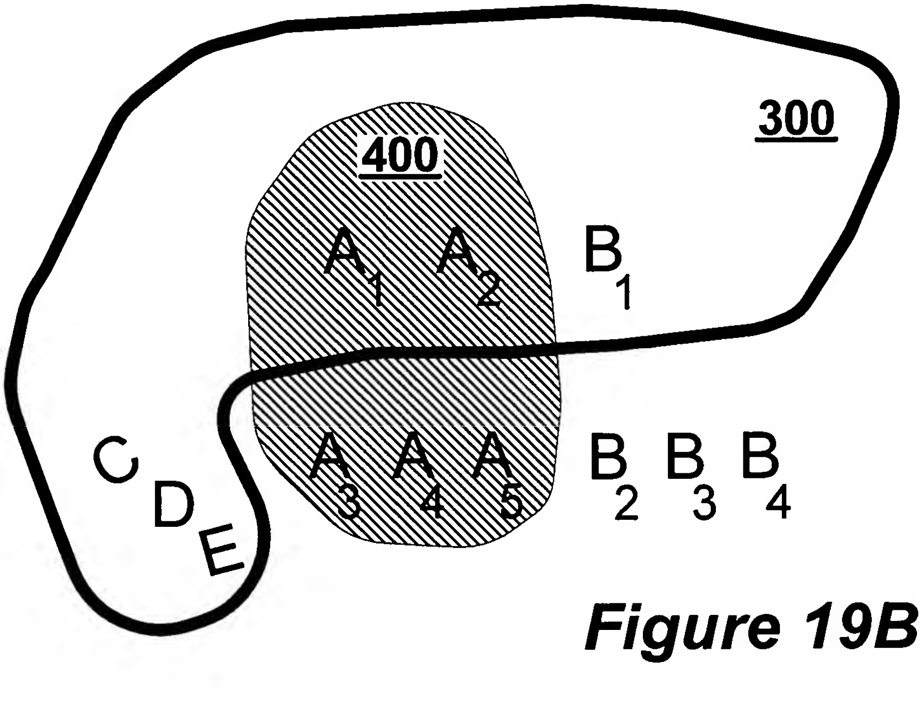

Figure 19B

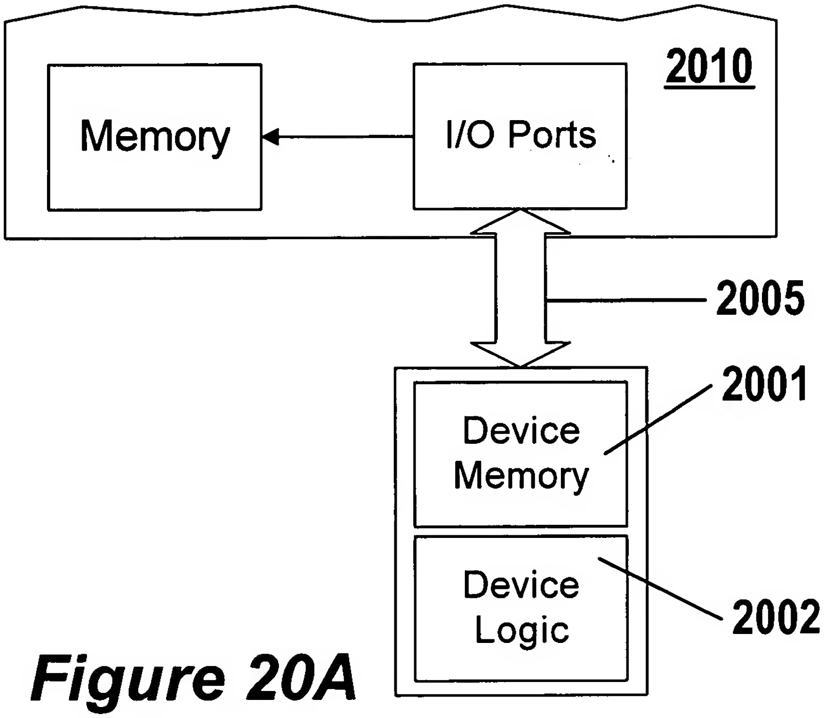

Figure 20A

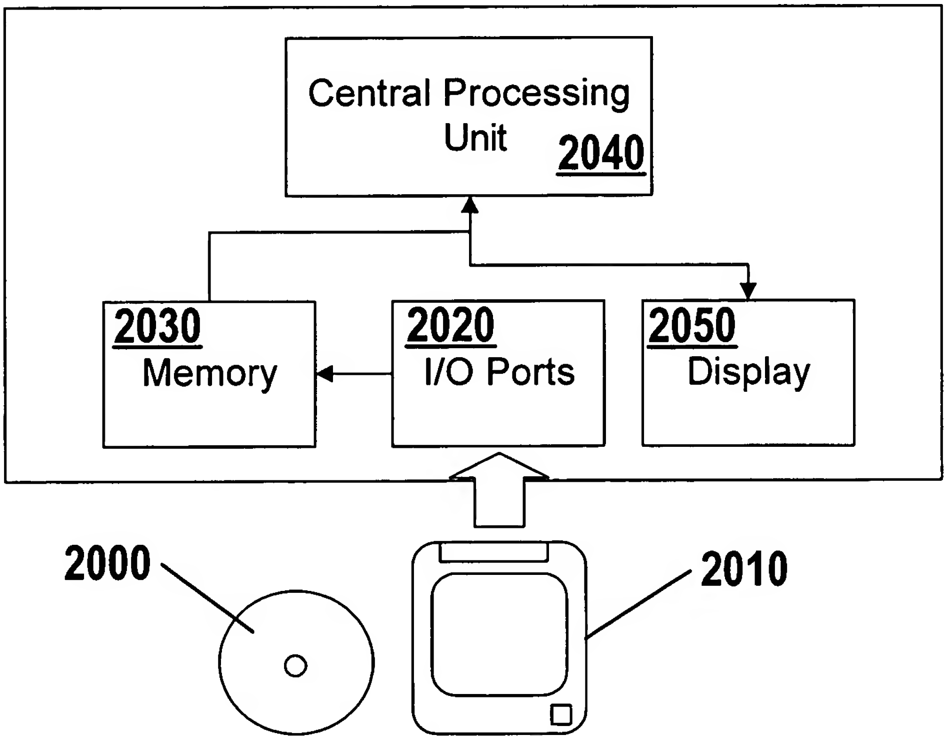

Figure 20B

```
PROGRAMNAME FOLLOWER
BEGIN
   ~COMMENTING Select the path
 2100  INPUT  N, Pathpoints
 2110  SET     Startpoint                          〈 Pathpoints(1)
 2120  SET     endpoint      〈         Pathpoints(N)
 2130  SET     Refpoint                                  〈       startpoint
 2140  SET     VoxelDistance     〈     Pathpoints(2)- Pathpoints(1)
   ~COMMENTING Select the geo-operator
 2200  INPUT  xlength, ylength, zlength
 2210  INPUT  Voxref_xyz
 2220  INPUT  Direction_xyz
 2230  INPUT  Algorithm
 2240  INPUT  ValidityInfringementRule
 2250  INPUT  K, DecisionBoundaryTable
 2260  SET     opx 〈   xlength
 2270  SET     opy 〈   ylength
 2280  SET     opz 〈   zlength

 2300      LOOPSET M   〈      N  ; DOWHILE M > 0
   ~COMMENTING Move the geo-operator if not at the startpoint
 2310          IF (Refpoint EQUAL startpoint)
 2320             SET Newpoint 〈   Refpoint
                OTHERWISE
 2330             SET Newpoint 〈    Refpoint + VoxelDistance
             ENDIF
 2340        CALCULATE    SmoothRotatedIndices(pathpoints Opx,Opy,Opz, ,Direction,Refpoint,Newpoint)
 2350        CALCULATE    VerticalFloat(pathpoints,Opx,Opy,Opz, Direction,Refpoint,Newpoint)
   ~COMMENTING Determine if valid computation is possible
 2400        CALCULATE    ValidityInfringementRule(SmoothRotatedIndices, VerticalFloat, Op_xyz, Newpoint)
 2410        IF (ValidityInfringementRule <= 0)
 2420              SET  VoxelValidityAttribute(Pathpoints(M)) 〈 FALSE
               OTHERWISE
 2430              SET  VoxelValidityAttribute(Pathpoints(M)) 〈  TRUE
 2440              LOOPSET H  〈    size(Opx,Opy,Opz)  ; DOWHILE H > 0
                      ~COMMENTING Calculate and store the geo-operator
 2500                 INPUT          BrickofData(SmoothRotatedIndices)
 2510                 CALCULATE   Algorithm(Pathpoint(M))
 2520                 STORE          CalculationResult(M)
 2530                 CALCULATE   DecisionBoundaryResult(DecisionBoundaryTable(CalculationResult(M)))
 2540                 SET              VoxelPathAttribute(Pathpoints(M))        〈    PathColor
 2560                 SET              VoxelCalculationAttribute(Pathpoints(M)) 〈
                      DecisionBoundaryResult(M)
 2570                 STORE        Voxel(...)Attribute(Pathpoints(M))
 2580                 DECREMENT  H
 2590             ENDOFLOOPON H
             ENDIF
   ~COMMENTING Increment the position and go to next path point
 2600        SET  Refpoint 〈 Refpoint + VoxelDistance
 2610        DECREMENT M
 2620      ENDOFLOOPON M
   ~COMMENTING Visualize results
 2700  RECALL  Voxel(...)Attribute(pathpoint(...))
 2710  DISPLAY Voxel(...)Attribute(pathpoint(...))
ENDPROGRAM.
```

Figure 21

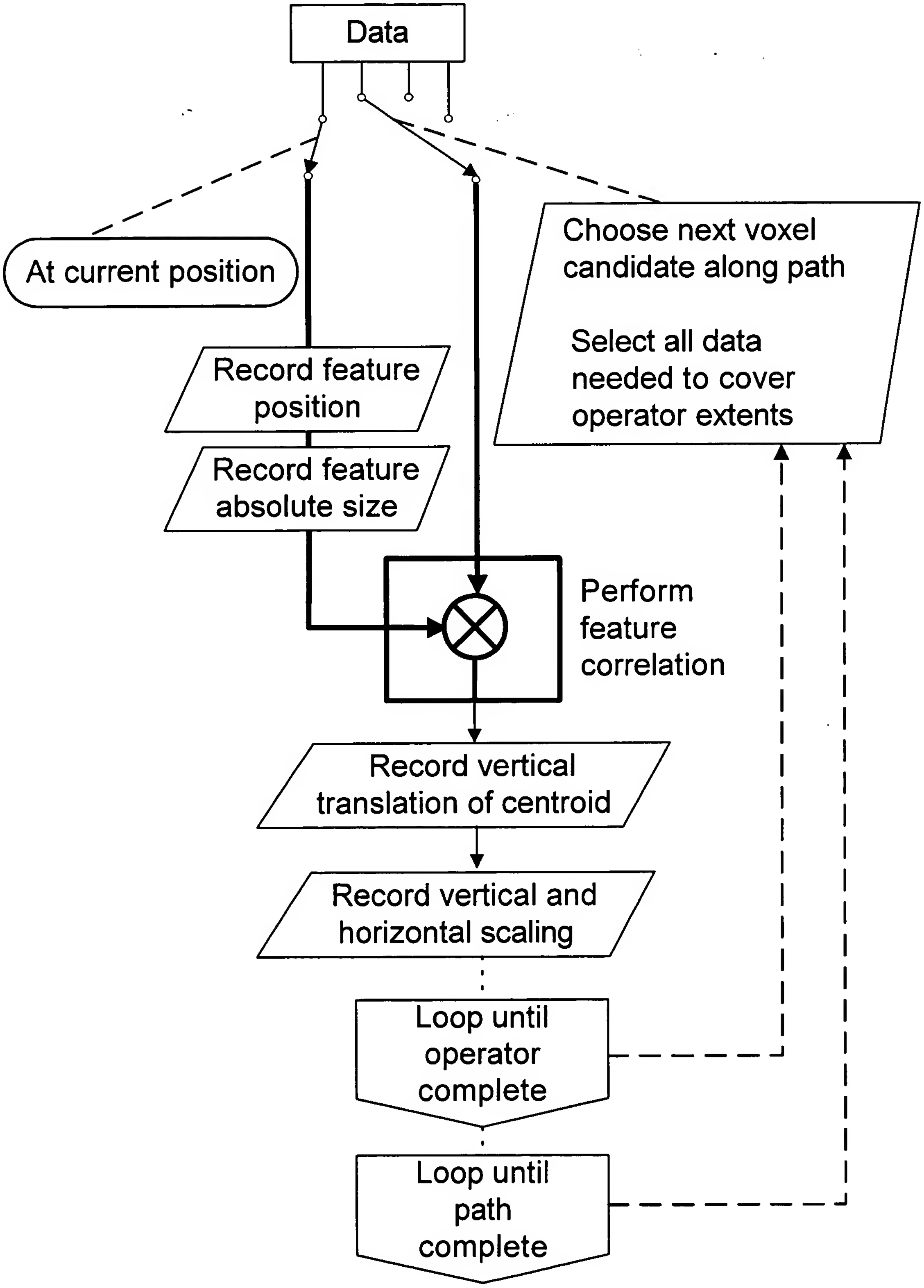
Data
At current position
Record feature position
Record feature absolute size
Choose next voxel candidate along path
Select all data needed to cover operator extents
Perform feature correlation
Record vertical translation of centroid
Record vertical and horizontal scaling
Loop until operator complete
Loop until path complete

Figure 22

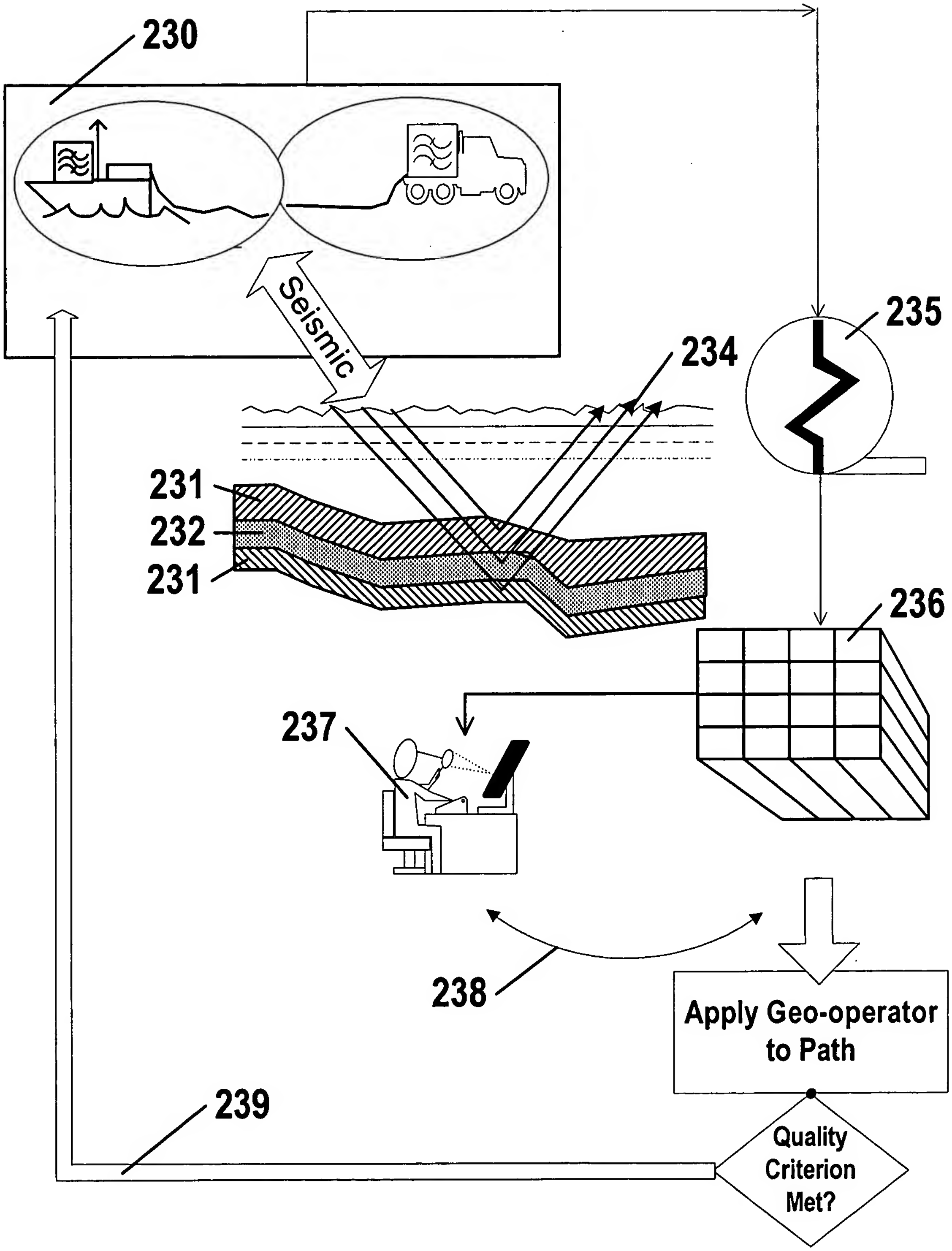
230
Seismic
234
235
231
232
231
236
237
238
Apply Geo-operator to Path
239
Quality Criterion Met?

Figure 23

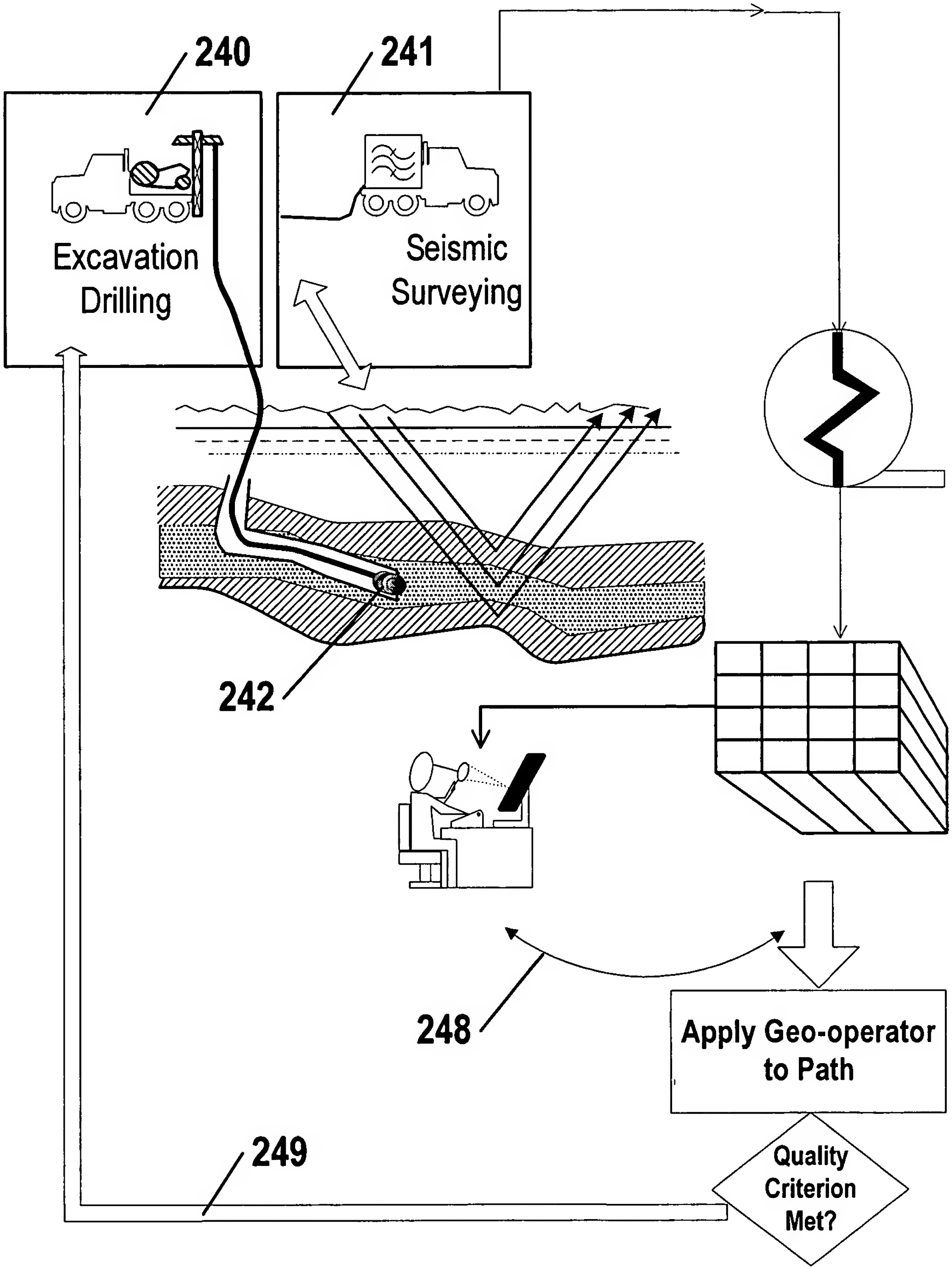
240
241
Excavation
Drilling
Seismic
Surveying
242
248
Apply Geo-operator
to Path
Quality
Criterion
Met?
249

Figure 24